

教育用基板マイクロコンピュータの設計(3)

川 端 信 賢*

Design of a One Board Microcomputer for Educational Use (3)

Shinken KAWABATA

In previously published papers^(11,12), we discussed the systems design of our microcomputer. This paper describes a portion of the hardware design of the computer. It involves the entire construction, the CPU group, the memory group and the hardware step control in the control group.

まえがき

前編^(11,12)までは本機の機能, 操作方法ならびに動作モードについて述べた。本文以降では本機のハードウェア構成について述べる。本文ではまずハードウェアの全体的構成, CPUグループ, メモリグループおよび制御グループ中のハードウェアステップ制御について述べる。本文の各番号は前編からの続き番号である。

5. 本機のハードウェア構成

まず, 5. 1で本機のハードウェアとソフトウェアの機能分担について述べたあと, 5. 2でハードウェアの設計目標および素子の選択方針を述べる。5. 3ではハードウェアを機能別にグループ分けして示す。5. 4以降ではハードウェアのグループ別にその機能, 構成ならびに動作仕様を説明する。

5. 1 ソフトウェアとハードウェアの機能分担

4. ままでに述べた本機の機能はハードウェアとソフトウェアにより実現されるわけであるが, 両者の機能分担について簡単に述べる。2. 1参照。なお, ハードウェアの詳しい仕様は5. 4以降で述べる。

(1) 16進表示器の表示

ハードウェアタイマからの周期的割込みにより,

表示プログラムを起動して, これにより表示を行わせる。

(2) キー入力 (Reset, Intrpt を除く)

他のワンボードコンピュータと同様に, キー入力部はエンコーダを持たない。代りにキースイッチの配列の縦線と横線の各線対に対して, 2線間の導通をソフトウェアにより高速検査する。これにより各線対に対応するキーの検出ならびに同定を行う。

(3) ハードウェアステップ

ステップ開始番地の設定から開始番地への飛越しまではソフトウェアによる。その後のステップ動作の制御はハードウェアによる。

(4) ソフトウェアステップ

ステップ開始番地の設定, 開始番地への飛越し, ステップ動作時のキー検出ならびにそれに伴う他モードへの移行など, このステップ動作の主な管理はソフトウェアにより行う。ただし, 1命令単位での実行停止ならびにこれに伴うモニタプログラムの呼出し (CALL) は割込み (ハードウェア) により行う。また, この割込み発生のタイミング制御は(3)の場合と同様, ソフトウェアの及ぶ範囲内ではソフトウェアにより行う。ソフトウェアの制御が及ばなくなった後の細かい制御はハードウェアによる。

(5) ラン&ブレーク

*管理工学教室

1982年5月31日受付

この動作の制御は(4)と同様に行う。

(6) インタラプトキー (Intrpt)

このキー入力1つの割込み要求入力であって、「割込み制御回路」を通じてCPUに受け付けられ、かつこの制御回路によりIntrptに対する割込み処理プログラムをCALLする。

(7) リセットキー (Reset)

このキー入力は「リセット回路」を通じてCPUに受け付けられ、0番地から始まる初期化プログラムに制御が引継がれる。

5.2 ハードウェアの設計目標

5.2.1 設計目標

本機のハードウェアの設計は次の(1)～(5)の目標に従った。

- (1) 教材用に適すること。
- (2) 素子数が少ないこと。
- (3) 消費電力が小さいこと。
- (4) 拡張性に富むこと。
- (5) 費用が小さいこと。

ここで(1)は本研究の基本目標であるが、この章では主にコンピュータのハードウェア教育ならびに基本的応用教育の観点から適することを意味する。具体的には本機のハードウェア構成の基礎的な部分が初心者に理解しやすく、かつ指導しやすい形になっていること。例えば、コンピュータのインタフェース回路およびその応用実験ははんだ付けやラッピングなしでも手軽に行えること。また、本機が教材用のコンピュータ応用制御装置などとして使えること。さらに、メモリチップ選択とI/Oポート選択に1組の回路(デコーダ)を兼用することもできるが⁽²⁾、これは初心者への説明用に適しないと考えられるので、本機では採用しない^{*1}。

(2) 素子数が少ないことは、同一機能でも必要なプリント基板の面積が小さくて済む。教材用として考えた場合、ハードウェア構成が見やすく、理解しやすくなる。組立て作業数が減る。信頼性の向上につながる。

(3) 消費電力が小さいことは、コンピュータ用の直流電源が小形・軽量で済むため、費用節減と同時に教室内外への携帯が容易になる。

(4) 拡張性に富むことは、本機を初歩的なコンピュータ教育用以外の目的に応用したい場合に必要と

なる。例えば、本機を開発用に使う場合にも拡張性が重要になる。市販の基板マイクロコンピュータには拡張性に欠ける機種が見られる⁽¹⁶⁾。

(5) 費用が小さいことは当然の目標であるが、特に教材用に必要な台数(多数)確保のために重要である。

5.2.2 使用素子の選択方針

前記の設計目標を達成するために、本機の設計ではCPUチップと同一ファミリー内の素子を原則的に採用している^(8,13)。同じ目的で次の①～③の場合にはこのファミリー以外の適切なTTL素子等を採用している^(14,15)。

① 必要な素子がCPUファミリー内に用意されていない。

② CPUファミリー内の同等素子よりも消費電力あるいは費用が小さく、かつ(動作速度、駆動能力および入力負荷の点でも)本機に適する素子が他にある。

③ CPUファミリー内の素子を使うよりも、素子数が少なくて済むような適切な素子が他にある。

この他に費用面では半導体製品の場合は、価格推移(一時的に指数関数的低下)を考慮して、現在の価格にあまりこだわらずに、数年先以降の価格を考えて選択する。大容量メモリチップなどの使用がその例である。

また、長期安定的に入手可能な部品であることも大事な条件であるが、これは小口利用者としてはメーカーとの長期供給契約を結ぶのが困難であり、長期保証は得られにくい。

5.3 本機のハードウェア構成

本機のハードウェアは次の各グループにより構成される。また、図11に本機のハードウェアのブロック図を示す。

ハードウェア構成要素

- (1) CPUグループ (図11の①, ②)
- (2) メモリグループ (同③, ④)
- (3) 制御グループ (同⑤～⑧)
- (4) 入出力グループ (同⑨～⑳)
- (5) パスライン他 (同㉔～㉚)

以下各グループ別に述べる。

5.4 CPUグループ

CPUグループの中心にはINTEL社の8ビット

*1 選択回路のこのような兼用は、別の目的のためにも望ましくない。つまり、I/Oポート番号の変化を引起すため、兼用では容量の異なるメモリチップへの自由な変更ができない。5.5.1参照。

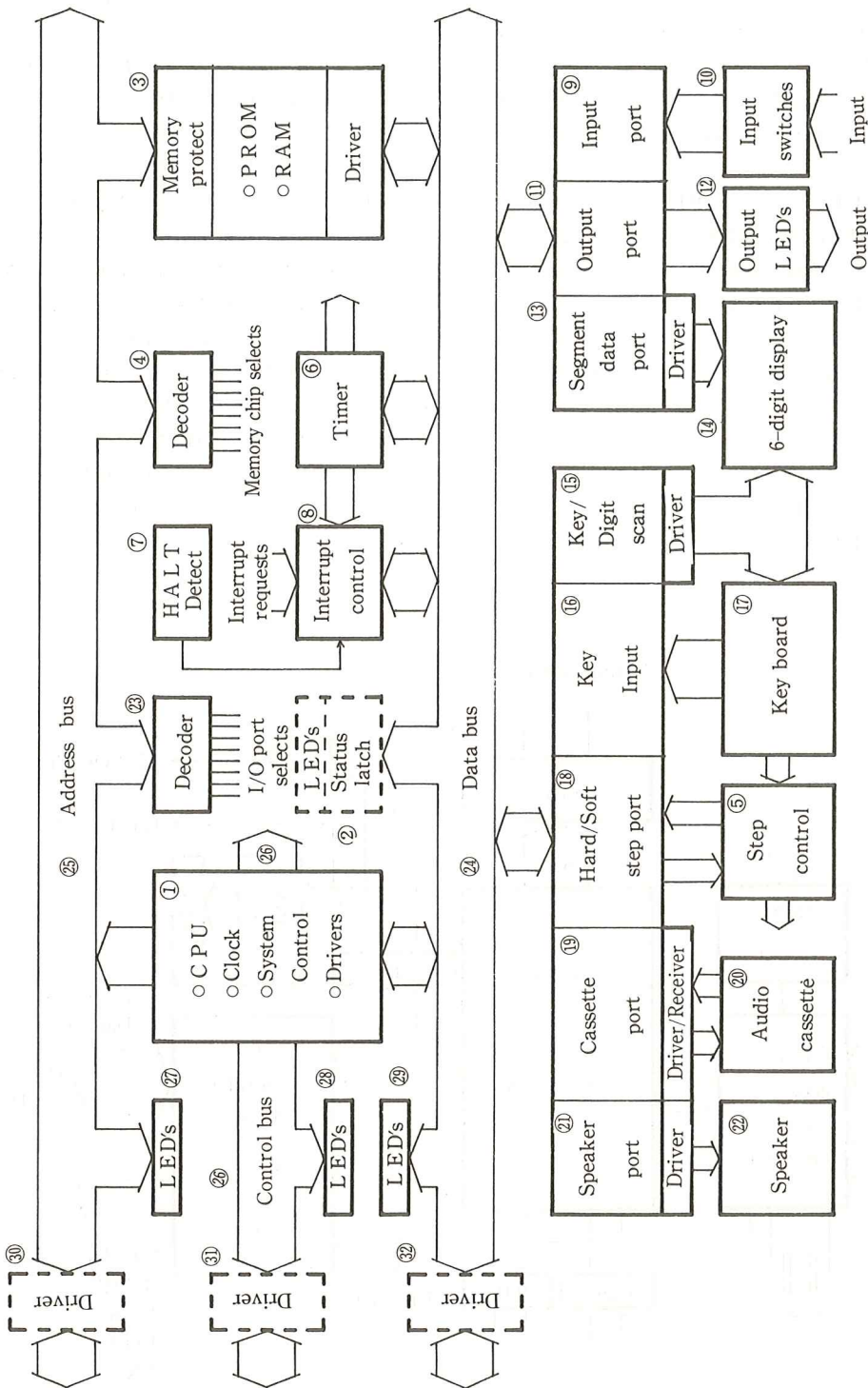


図11 教育用基板マイクロコンピュータのブロック図

マイクロプロセッサ 8080A ファミリーより 8080A, 8224, 8228/8238 またはこれらの相当品を採用する。したがって、CPU グループ全体としての機能および仕様の基本は、これらのファミリー素子によりおのずと決まる。これらの素子の詳細な機能および仕様については技術資料^(7,8,9,13)を参照されたい。

構成 CPU グループは CPU チップと、このチップの機能を補い、あるいは強化するための回路により構成される。主な構成要素は次のとおりである。

- (1) CPU (8080A マイクロプロセッサ)
- (2) クロックジェネレータ/ドライバ (8224)
- (3) システムコントローラ/ドライバ (8228 または 8238)
- (4) バスドライバ
- (5) ステータスラッチ
- (6) リセット回路

図12に CPU グループのブロック図を示す。

機能/仕様 CPU グループの機能は述べるまでもないが、簡単に言えばメモリに書込まれた CPU の命令 (プログラム) に従って、計算その他の処理動作を行い、かつ外部からの幾つかの要求 (リセット、ホールド、ウェイト、割込み) に対して、それぞれハードウェア的に定められた特別な動作を行うことである。

5.4.1 CPUバスの構成

CPU グループと外部との情報交換を行うための主な信号の種類は次のとおりである。したがって、CPU グループ内部だけで用いる信号等はこれに含まれない。カッコ内はそれぞれ信号の記号名、入出力の別およびこの信号に主に関係する CPU グループ内チップの型番を示し、横線 (バー) の付いた信号は反転形 (いわゆるアクティブ ロウ) である。

① データバス (DB₇—DB₀;

入力/出力; 8228/8080A)

② アドレスバス (A₁₅—A₀; 出力; 8080A)

③ メモリリード信号

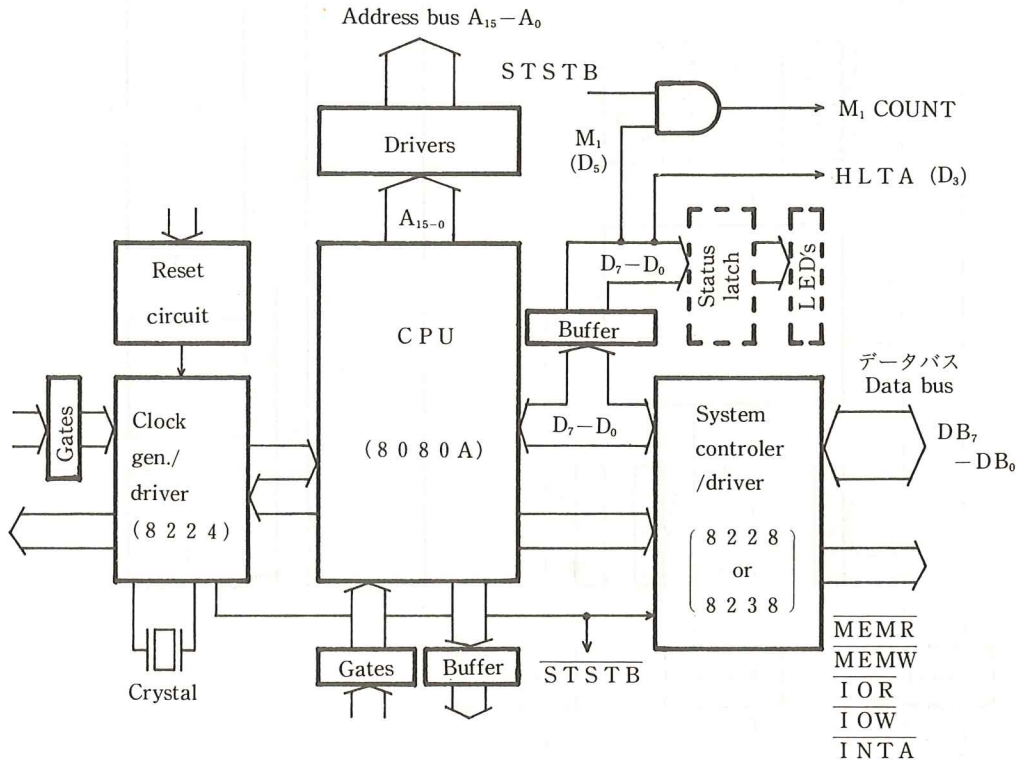


図12 CPUグループのブロック図

- (MEMR; 出力; 8228/8080A)
- ④メモリライト信号
(MEMW; 出力; 8228/8080A)
- ⑤I/Oリード信号
(IOR; 出力; 8228/8080A)
- ⑥I/Oライト信号
(IOW; 出力; 8228/8080A)
- ⑦リセット入力信号
(RESIN; 入力; 8224/8080A)
- ⑧リセット信号
(RESET; 出力; 8224/8080A)
- ⑨ウェイト要求信号
(WAITRQ; 入力; 8224/8080A)
- ⑩ウェイト応答信号
(WAIT; 出力; 8080A)
- ⑪割込み許可信号 (INTE; 出力; 8080A)
- ⑫割込み要求信号 (INT; 入力; 8080A)
- ⑬割込み応答信号
(INTA; 出力; 8228/8080A)
- ⑭ホールド要求信号
(HOLD; 入力; 8080A)
- ⑮ホールド応答信号
(HLDA; 出力; 8080A)
- ⑯OSCクロック信号 (OSC; 出力; 8224)
- ⑰ ϕ_2 クロック信号
(ϕ_2 (TTL); 出力; 8224)
- ⑱SYNC同期信号
(SYNC; 出力; 8080A)
- ⑲ステータスストローブ信号
(STSTB; 出力; 8224/8080A)
- ⑳M₁カウント用信号
(M₁COUNT; 出力; 8080A/8224)
- ㉑ホールド応答信号
(HLTA; 出力; 8080A)

ここで、③～㉑をまとめてコントロール信号あるいはコントロールバスと呼ぶ。

5. 4. 2 CPUバスの機能と使用法

①データバス (DB₇—DB₀)

データバス (DB₇—DB₀) は基本的にはCPUとメモリまたは入出力間で命令あるいはデータの授受を行うためのデータ用信号線である。すなわち、CPUが

らメモリや出力ポートにデータを格納あるいは出力する場合に、CPUはこのデータをデータバスに送り出す(乗せる)。②、④、⑥参照。逆に、メモリや入力ポートから命令あるいはデータをCPUに読込む場合は、CPUからの制御信号(③、⑤参照)によりメモリや入力ポートからデータバスに乗せられたデータをCPUに読込む。

このほかメモリと入出力機器間でデータの直接転送(いわゆるDMA転送)を行うためにも、データバス(および以下に述べる2つのバス)が用いられる*1。

②アドレスバス (A₁₅—A₀)

アドレスバスはメモリ番地(アドレス)指定信号(A₁₅—A₀)および入出力ポート指定信号(A₇—A₀)またはA₁₅—A₈)を兼ねる16本の信号線である。すなわち、CPUがメモリからデータを読込む場合、読込むべきデータが格納されているメモリチップおよびそのチップ内の番地を指定するための16ビット信号を、CPUはこのアドレスバスに出す。2¹⁶=65,536であるから、最大65,536番地を直接指定できる。

一般に、アドレスバスの上位ビット、(A₁₅, A₁₄, ..., A_i)はチップ選択用に、また下位ビット(A_{i-1}, ..., A₁, A₀)は選択されたチップ内の番地選択用に用いられる。図13参照。アドレスバスのチップ選択用と番地選択用の境界ビットA_iは、使用するメモリチップの語数により決まる。例えば、チップの記憶構成が2048語×8ビットであるメモリチップを使用する場合、2048=2¹¹であるから、i=11である。

次に、CPUが入力ポートからデータを読込む場合、(複数の入力ポートの中から)特定の入力ポートを指定するための8ビット信号をアドレスバスの下位8ビット(A₇—A₀)に出す。(この信号は同時に上位8ビット(A₁₅—A₈)にも出される。)この場合もメモリ番地選択の場合と同様、一般に入出力(I/O)アドレスの上位ビット(A₇, A₆, ..., A_j)はI/Oポート選択回路(デコーダ)自身の選択用に用いられ、下位ビット(A_{j-1}, ..., A₁, A₀)はこの選択回路により選択されるべき2^j個のI/Oポートの選択用に用いられる。2⁸=256であるから、最大256個の入力ポートおよび出力ポートが直接指定できる。図14参照。

このほか、メモリへのデータ書込みおよび出力ポートへのデータ出力の場合も、同様のアドレス信号がC

*1 本機は他のワンボードマイクロコンピュータと同様に、DMAコントロール回路を内蔵しないが、このコントロール回路を本機の外部に取付けることにより、DMA転送の実験が可能のように、必要な信号線を用意する。⑭、⑮参照。

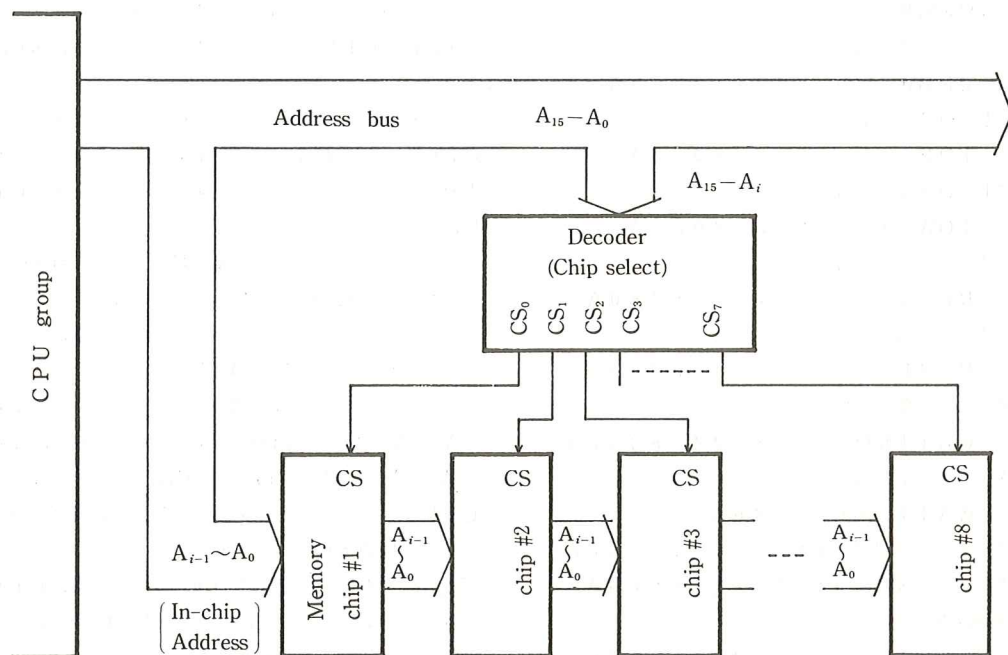


図13 アドレスバスとメモリ番地選択

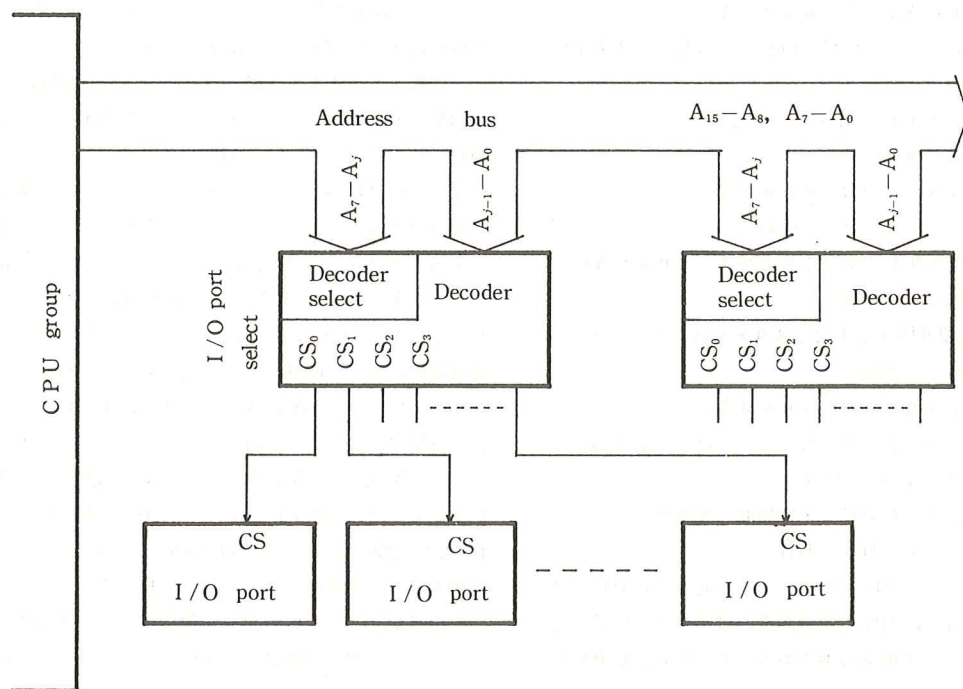


図14 アドレスバスとI/Oポート選択

PUから出される。

③メモリリード信号 (MEMR)

これはメモリデータの読出しをコントロール、あるいは知らせるための信号である。CPUがメモリデータを読出すときには、CPUはこの信号を低(ロウ)レベルにし、その他のときは高(ハイ)レベルまたは高インピーダンス状態にする。メモリ回路はこの信号がロウレベルのときにのみ、メモリデータをデータバスに出す(乗せる)ように設計しなければならない。

④メモリライト信号 (MEMW)

これはメモリへのデータ書き込みをコントロールする信号である。CPUからメモリにデータを書込むときには、CPUはこの信号をロウレベルにし、その他のときは、ハイレベルまたは高インピーダンス状態にする。メモリ回路はこの信号がロウレベルのときにのみ、データバス上のデータをメモリに書き込むように設計しなければならない。

⑤I/Oリード信号 (IOR)

入出力(I/O; Input/Output)コントロール信号の1つであり、入力ポートあるいは入力機器からのデータ入力をコントロールする。本機もいわゆるアイソレーテッドI/O方式を採用しているから、CPUが入力データを読込むとき(すなわち、IN命令実行時)には、CPUはこの信号をロウレベルにし、その他のときはハイレベルまたは高インピーダンス状態にする。したがって、入力ポートはこの信号がロウレベルのときにのみ、入力データをデータバス上に乗せるように設計しなければならない。

⑥I/Oライト信号 (IOW)

入出力コントロール信号の1つであり、出力ポートあるいは出力機器へのデータの出力をコントロールする。アイソレーテッドI/O方式であるため、CPUがデータを出力(OUT命令実行)するときは、CPUはこのコントロール信号をロウレベルにし、その他のときはハイレベルまたは高インピーダンス状態にする。したがって、出力ポートはこの信号がロウレベルのときにのみ、データバス上のデータ(=CPUからの出力データ)を取込むように設計すればよい。

⑦リセットイン信号 (RES IN)

CPUその他の回路をリセット(初期化)するための原入力信号である。実際にはこれがCPUグループ内で整形・調整されてから用いられる。⑧参照。この

入力に対しては(8224内に)シュミット回路が内蔵されているため、入力信号は立上り立下り時間の長い波形でもかまわない。したがって簡単なRC充放電回路によるパワーオンリセットおよび手動リセットが可能である。

⑧リセット信号 (RESET)

⑦リセットイン信号がCPUグループ内(8224)で波形整形およびタイミング調整されてこのリセット信号となる。8080A・CPUその他の回路のリセット信号として用いられる。リセット信号は少くともCPUの3クロック間持続する必要がある⁽¹³⁾。リセット回路の設計にはこの点を考慮する。

⑨ウェイト要求信号 (WAITRQまたはRDYIN)

CPUと外部(低速メモリや入出力機器)間のデータ授受の時間調整を行うために用意された信号である。この信号のハイレベルは本来メモリや入出力機器がCPUに対して、データ授受の用意ができていないこと(READYである)ことを伝えるための信号である。逆にロウレベルはデータ授受の用意ができていないこと、したがって、外部からCPUに対して“待ち”(ウェイト, WAIT)状態に入るよう要求するための信号である。この信号はCPUグループ内の8224でタイミング調整されたあと、CPUのREADY端子に加えられる。

CPUは各マシンサイクル $M_i = (T_1 T_2 T_3 (T_4 T_5))^* 1$, ($i = 1 \sim 5$)の T_2 ステートでREADY入力を検出し、この結果 $READY = 0$ ならば、この間CPUは T_2 ステートをくり返す。すなわち、これがウェイト状態であって、このときの T_2 ステートのことをとくにウェイト(WAIT)の意味を込めて T_w ステートと呼ぶ。 T_2 または T_w ステートで $READY = 1$ を検出した場合は、CPUは T_3 ステートに進む。⑩参照。

ハードウェアステップ動作は、このウェイト要求入力を制御することにより行わせる。5. 6. 1参照。

⑩ウェイト応答信号 (WAIT)

⑨ウェイト要求信号に対するCPUの応答信号である。マシンサイクルの T_2 ステートで $READY = 0$ のとき、CPUは $WAIT = 1$ を出力し、ウェイト状態にあることを外部に対して知らせる。このほか、CPUがHALT状態(T_{wh} ステート)に入った場合に

*1 8080Aの基本的マシンサイクルは T_3 ステートまでにより構成されるが、その他に T_4 あるいは T_5 ステートまで必要とするものもある。⁽⁷⁾

もWAIT=1となる。

⑪割込み許可信号 (INTE)

CPU内部の割込み許可フリップフロップ (INTE F/F) の状態を表す信号である。すなわち、CPUが割込み受け可能な状態にあるか否かを外部に知らせる信号である。INTE=1のとき割込み可能、INTE=0のとき不可能である。

このフリップフロップはEI命令およびDI命令によって、それぞれセット(1)およびリセット(0)される。また、割込みが受け付けられた直後のT₁ステートで自動的にリセットされる。さらに、⑧リセット信号によってもリセットされる。8080Aの割込みを利用する場合、プログラミング上これらの点に注意を要する*¹。⑫、⑬参照。

⑫割込み要求信号 (INT)

これはCPUに対する割込み要求信号であり、実行中の命令の終了時、あるいはHALT中に検出される。ただし、CPUがホールドのときまたは割込み許可フリップフロップ (INTE F/F) がリセットされているときには、割込み要求は受け付けられない。⑪、⑬参照。

⑬割込み応答信号 (INTA)

⑫割込み要求信号に対するCPUの応答信号である。割込み要求を受け付けたとき、この信号は次のマシンサイクルM₁のT₂からT₃ステートにかけてロウレベルになる。このときのデータバス上にあるデータをCPUは割込み制御回路から与えられた命令として読む。したがって、割込み制御回路はこのINTA信号がロウレベルのときに、データバス上に必要な命令を乗せるように設計しなければならない*²。

⑭ホールド要求信号 (HOLD)

CPUがホールド状態 (⑮参照) に入るよう外部から要求するための信号である。この信号は次の条件下で受け付けられる。

- CPUがHALT状態T_{WH}であるとき、または
- CPUがT₂あるいはT_WステートにおいてREADY=1であるとき*³。

HOLD信号のタイミングについては⁽¹⁸⁾、

④HOLDの立上りは ϕ_2 の立上り前にセットアップ時間 (t_{HS} =最小140ns) が必要。

⑤HOLDの立下りは、 ϕ_2 の立上り後、保持時間 (t_H =最小0 ns) が必要。

ホールド要求信号は⑮ホールド応答信号とともに、DMA転送に用いられる⁽¹⁷⁾。

⑮ホールド応答信号 (HLDA)

この信号は⑭ホールド要求信号に対するCPUからの応答信号であり、ホールド要求が受け付けられたときにHLDA=1になる。このとき、CPUの動作は進行中のマシンサイクル終了後中断し、8080Aのアドレスバス (A₁₅-A₀) およびデータバス (D₇-D₀) が高インピーダンス状態になる*⁴。また、このHLDA信号をバスドライバおよび8228/8238の3状態ゲート制御入力 (\overline{G} および \overline{BUSEN}) に結線することにより、HLDA=1のときシステムアドレスバス (A₁₅-A₀)、8228から出るシステムデータバス (DB₇-DB₀) および一部のコントロール信号 (\overline{MEMR} , \overline{MEMW} , \overline{IOR} , \overline{IOW} , \overline{INTA}) も同時に高インピーダンス状態になる。図12参照。

HLDA信号はDMA転送等に用いる。

⑯OSC信号および⑰ ϕ_2 (TTL) 信号

これらはTTLレベルのクロック信号であり、CPU以外の回路の (CPUに対する) タイミング用に用いることができる。本機では ϕ_2 (TTL) をタイマ用の1つのクロック入力として用いる。

⑱SYNC同期信号 (SYNC)

8080Aの各マシンサイクルのはじめに出力される信号である。このとき、データバス (D₇-D₀) にはステータス情報が出力される。⑲参照。本機ではSYNC信号をステップ制御回路で、マシンサイクル数のカウント用パルスとして用いる。

⑲ステータスストロブ信号 (\overline{STSTB})

8080Aの各マシンサイクルのはじめに、そのデータバス (D₇-D₀) に一時的に出力されるCPUのステータス情報をレジスタ (ステータスラッチ回路)

*1 これに対して、8085A^(8,13)やZ-80A⁽¹⁹⁾マイクロプロセッサ等は、割込み許可フリップフロップの影響を受けない優先割込み入力をもつ。

*2 8228/8238によりこのINTA信号は割込み制御回路から複数バイト命令が与えられる場合でも、その各バイトの読込みに必要な回数だけ発生する。すなわち、CALL命令 (3バイト) に対してM₁~M₃サイクルにわたって3回のINTA信号が発生する。また、INTA信号の発生タイミングはIORおよびMEMR信号と同じである。

*3 ただし、DAD命令のM₂, M₃サイクルの場合は、例外的にREADY=0のときでもHOLD要求を受け付ける。

*4 CPUのこのような状態をホールド状態という。

に取込む(ラッチする)ためのストロブ信号である。ラッチされたステータス情報はコントロール信号の合成のために用いられる。本機ではこのステータス情報をLEDにより表示させ、ハードウェアステップ動作時に観察させる。

②M₁ カウント用信号 (M₁ COUNT)

第1マシンサイクル(M₁)中に1個発生する正のパルス信号である。(D₅に現われるM₁ステータス信号とSTSTB信号の論理積で作った信号。)本機ではこの信号をステップ制御回路において、命令サイクルのカウント用に用いる。

②ホールト応答信号 (HLTA)

8080AがHLT命令を実行すると、D₃にステータス情報としてHLTA=1を出力して停止する。本機ではホールト検出回路においてこの信号を検出したとき、CPUに対する割込み信号を発生できる。

タス情報としてHLTA=1を出力して停止する。本機ではホールト検出回路においてこの信号を検出したとき、CPUに対する割込み信号を発生できる。

5.5 メモリグループ

機能 プログラムやデータの半永久的あるいは一時的記憶機能をもつ。ある領域に対する書き込み保護機能ならびに主電源を切った場合のデータ保持機能をもつ。

機成 メモリグループは次の各要素により構成される。このブロック図を図15に示す。

- (1) 読み出し専用メモリ (ROM)
- (2) 読み書き可能メモリ (RWMまたはRAM)
- (3) メモリチップ選択回路

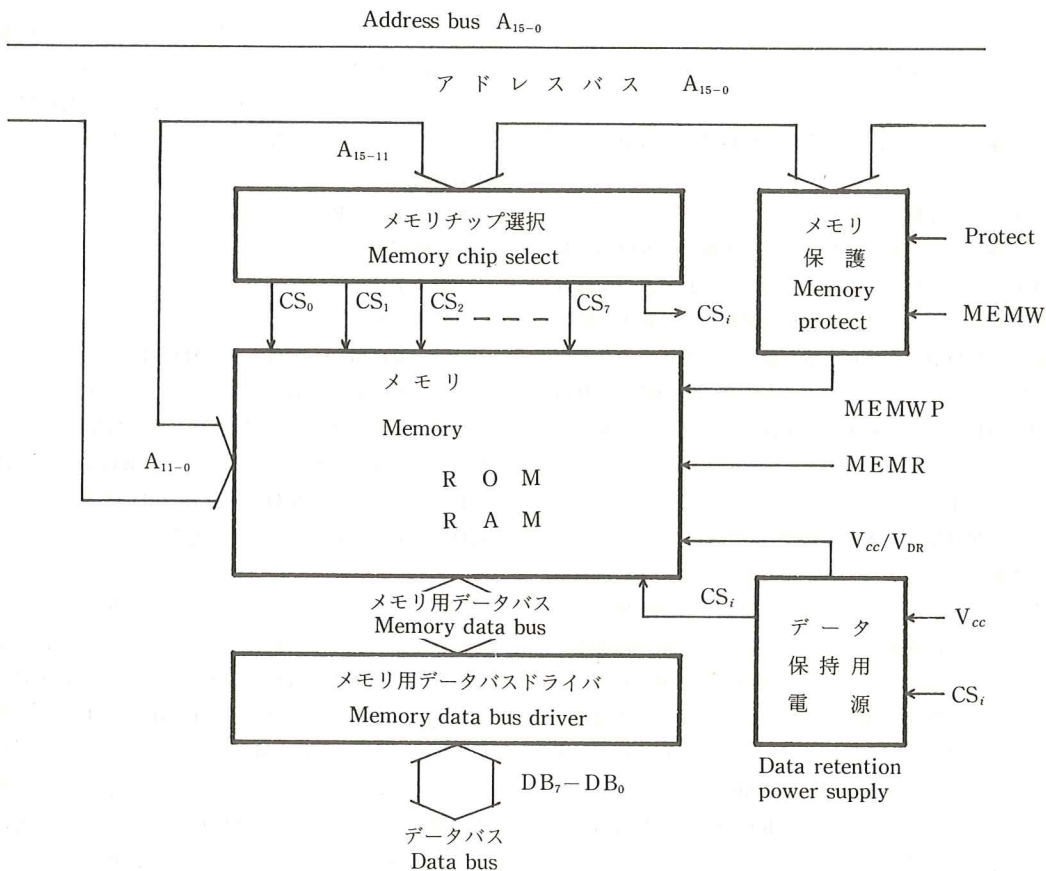


図15 メモリグループのブロック図

表3 本機の記憶容量 (単位Kバイト)

| 内 訳 | | ROM | RAM | 合 計 | 増減単位 |
|-----|------------------|-----|-----|--------|---------|
| 基板上 | 最大* ¹ | 14 | 14 | 16 | 2 |
| | 標準* ² | 4 | 4 | 8(6) | 4 |
| | 最大* ³ | 28 | 28 | 32(30) | 4 |
| 基板外 | 可 能 容 量 | | | 32 | 未 定 |
| 合 計 | 可 能 容 量 | | | 64(62) | 2, 4 ほか |

* 1 16Kビットチップ使用時の記憶容量。

* 2, 3 32Kビットチップ使用時の記憶容量。
32KビットPROMチップと同ビット数で、かつこれとピン互換性のあるスタティックRAMチップが将来入手可能になることを仮定している。当面これを16KビットRAMチップで代用するため、この間* 2, 3のRAM容量は表3の半分である。()内数字はRAM容量を2Kバイト(16Kビットチップ1個)にした場合の記憶容量である。

- (4) メモリ用データバスバッファ／ドライバ
(5) メモリ保護回路
(6) データ保持用電源回路

仕様 ここでは、記憶容量、ROMとRAMの差替えならびに使用記憶素子について述べる。

(1) 記憶容量

本機の記憶容量を表3に示す。本機ではROMとRAMの差替えが容易に行えるため、それぞれの記憶容量は可変となっている。図16に基板上での最大記憶容量時のROMおよびRAM容量の内訳を7とおり示す。一例として同図⑦は4Kバイトチップ使用時に、ROMが28Kバイト、RAMが4Kバイト、合計32Kバイトであることを示す。

(2) 専用コンピュータ化のための

ROMとRAMの差替え

本機をコンピュータ教育用の教材として用いる場合、単にソフトウェアおよびハードウェアの訓練用としてだけでなく、簡単な専用コンピュータあるいは専用コントローラのモデルとしても使用する計画である。このために本機は訓練用あるいは開発用から、専用機に容易に切替えられなければならない。これを可能とするために本機ではROMとRAMの差替えができるようにする。具体的に言えば、ROMおよびRAMチップは互いに差替えの容易な「ピン互換性」のあるものを選ぶ。プリント基板上ではメモリチップ領域にROM、RAM両用の信号線をプリント配線しておき、ROMとRAMで異なる一部の信号線は、はんだまたは

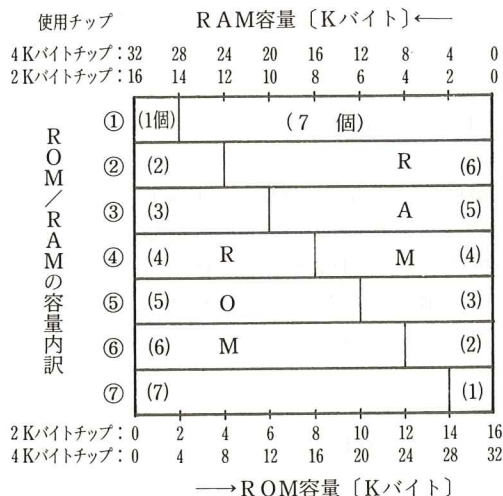


図16 基板上最大記憶容量時のROM/RAM内訳
(7通り×使用チップの容量2種類、
()内はチップ数)

ジャンパ線によりそれぞれ選択して結線する。図17参照。図18に本機の基板上のメモリチップの番地割付けを示す。0番地から始まる最初の4K番地はROM領域であり、その次の4K番地(または2K番地)はRAM領域である。以後は4K(または2K)番地単位でROMまたはRAMを増減できる。

(3) 使用記憶素子

記憶素子としてはMOSメモリを用いる。

使用記憶素子

ROM……MOS・PROM
RAM……MOS・スタティックRAM

本機の目的は教育用であるから、ROM内容の追加・修正が予想される。したがって、ROMとしては、これが容易に可能なMOS・PROMが適している。MOS・PROMは現在、性能価格比からも適切である。

また、同じ目的でCPUのウェイト状態を利用してハードウェアステップ動作を行わせるために、RAMとしてはMOSまたはCMOSスタティックRAMが適している。データ保持用にはCMOS・RAMを用いる。ダイナミックRAMは数ミリ秒毎に再書き込み(リフレッシュ)が必要なため、ハードウェアステップ動作に適しない。また、PROMとのピン互換性に欠ける^(13,20,21)。次に、バイポーラメモリは消費電力が大きく、チップ当りの記憶容量が小さい。しかも、バイポーラPROMはデータの消去・修正ができない^(13,20)。以上の理由により、ダイナミックRAMおよびバイポーラ

図17 本機に使用可能なメモリチップのピン配置とジャンパ配置例(13,20,21,22)

| Address | | | |
|-------------------------|-------------------------|---------------|---------------------|
| Decimal | Hexadecimal | (a) | (b) |
| 10進番地 | 16進番地 | 4 Kバイトチップ使用時 | 2 Kバイトチップ使用時 |
| 0 ₍₁₀₎ | 0 0 0 0 ₍₁₆₎ | チップ No. 1 | チップ No. 1 |
| | | R O M | R O M |
| 2 0 4 8 ₍₁₀₎ | 0 8 0 0 ₍₁₆₎ | | No. 2 |
| | | | R O M |
| 4 0 9 6 ₍₁₀₎ | 1 0 0 0 ₍₁₆₎ | No. 2 | No. 3 |
| | | R A M | R A M |
| 6 1 4 4 | 1 8 0 0 | | (No. 4) |
| | | | (ROM / RAM) |
| 8 1 9 2 | 2 0 0 0 | (No. 3) | (No. 5) |
| | | (ROM / RAM) | (ROM / RAM) |
| 1 0 2 4 0 | 2 8 0 0 | | (No. 6) |
| | | | (ROM / RAM) |
| 1 2 2 8 8 | 3 0 0 0 | (No. 4) | (No. 7) |
| | | (ROM / RAM) | (ROM / RAM) |
| 1 4 3 3 6 | 3 8 0 0 | | (ROM / RAM No. 8) |
| 1 6 3 8 3 | 3 F F F ← | | 基板上最大番地 |
| 1 6 3 8 4 | 4 0 0 0 | (No. 5) | () 内はオプション |
| | | (ROM / RAM) | |
| 2 0 4 8 0 | 5 0 0 0 | (No. 6) | |
| | | (ROM / RAM) | |
| 2 4 5 7 6 | 6 0 0 0 | (No. 7) | |
| | | (ROM / RAM) | |
| 2 8 6 7 2 | 7 0 0 0 | (No. 8) | |
| | | (ROM / RAM) | |
| 3 2 7 6 7 | 7 F F F ← | 基板上最大番地 | |
| 3 2 7 6 8 | 8 0 0 0 | | |

図18 基板上のメモリチップの番地割付け

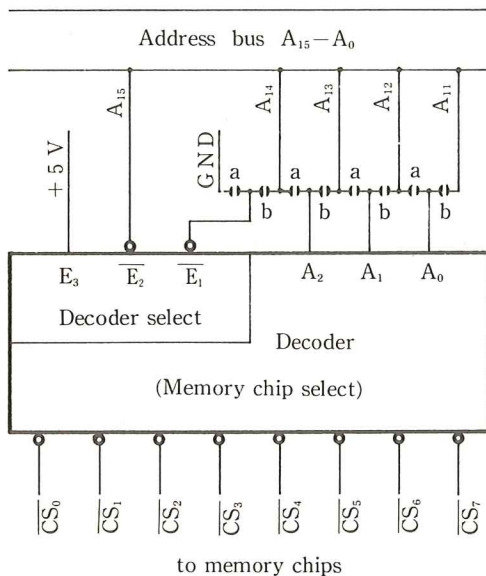


図19 メモリチップ選択回路

〔 4 Kバイトチップ使用時: a 点を接続〕
〔 2 Kバイトチップ使用時: b 点を接続〕

ラメモリは本機に適しない。

次にメモリグループの各構成要素について述べる。
ここで、(1)読出し専用メモリ (ROM)、(2)読み書き可能メモリ (RAM) については、すでにメモリの仕様の中で述べたので、これ以上の説明は不要であろう。
(3)メモリチップ選択回路ほかについて述べる。

5. 5. 1 メモリチップ選択回路

機能 メモリは一般に複数個のメモリチップを用いて、必要な大きさの記憶容量を構成する。メモリチップ選択回路はアドレス信号により指定されたメモリ番地に対応する一組のメモリチップの選択を行い、そのメモリチップへのデータの書き込みまたはデータの読出しを可能にする。

構成 アドレス信号の上位ビットを入力とし、チップ選択信号を出力にもつ一種のデコードである。図19および5. 4. 2-②アドレスバス参照。

仕様 一組のチップ選択回路が4 Kバイトチップ選択用と2 Kバイトチップ選択用に相互切替えが可能となるよう、この選択回路のアドレス入力の一部の結線を、はんだまたはジャバラ線により選択できる方式とする。図19参照。

5. 5. 2 メモリ用データバスバッファ／ドライバ

機能 ①データバスのメモリ側とそれ以外の部分の

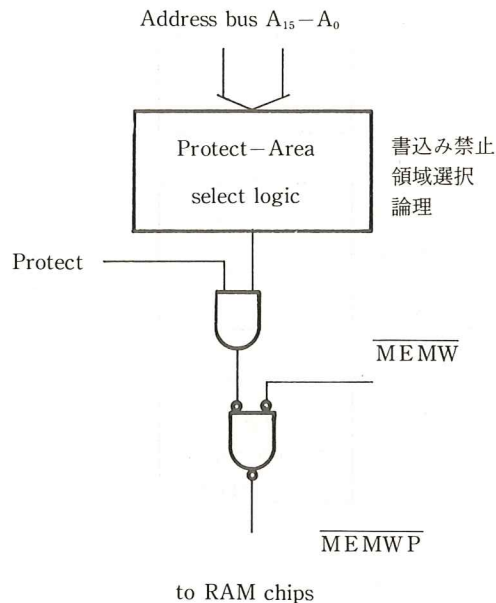


図20 メモリ保護回路の構成

分離による雑音しゃ断

②データバスの交流特性改善

③同上直流特性改善

必要性 ①および②はいずれも本機の記憶容量が大きく、したがってプリント基板上メモリ領域でのデータバス等の配線長が長いために必要となる。すなわち、データバスはメモリ部のほかに入出力ポート等周辺回路部への延長も必要であるため、信号線の全長が長くなり、そのインダクタンスおよび静電容量が大きくなる。このため④信号間の誘導雑音が大きくなり、⑤信号波形 (立上り立下り等交流特性) の劣化を生じる。加えてデータバスはアドレスバスに比べてタイミング上の余裕が小さい。

③は教育的観点からデータバスに信号表示用LEDを直結するために必要である。すなわち、MOS・LSIメモリの出力シンク電流 (2 mA程度^(13,20,21)) は、システムコントローラ/バスドライバの場合 (10 mA程度^(8,9,13)) と異なり、LED駆動電流 (少くとも3 mA前後必要) を流し込むには小さすぎる。

データバスのCPUおよび入出力ポート側とメモリ側の間に両方向性バスバッファ、すなわちメモリ用データバスドライバを入れることにより、上記①、②、③の解決をはかる。

5. 6. 3 メモリ保護回路

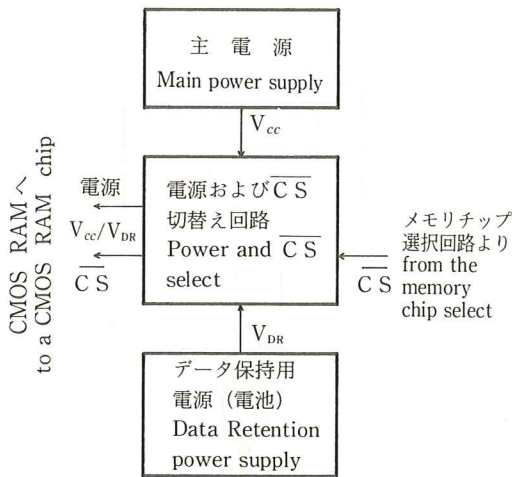


図21 データ保持用電源回路構成図

機能 RAMの特定番地領域に対して、不測の誤った書き込みを防止する機能をもつ。

これはプログラムの暴走による、RAM上のプログラムやデータの破壊を防止するためである。勿論、必要に応じてプログラムによりこの書き込み禁止を解除できるものとする。図20にメモリ保護回路の構成図を示す。

5. 5. 4 データ保持用電源回路

機能 コンピュータの主電源を切った場合に、特定のCMOS・RAM上のデータを保持する機能をもつ。

構成 次の(1)、(2)により構成される。図21参照。

- (1) データ保持用電源（電池）
- (2) 電源（およびCS信号）切替え回路

仕様 (2)電源切替え回路の入力は①主電源、②データ保持用電源および③CMOS・RAM選択用CS信号である。出力は①CMOS・RAM用電源および②同CS信号である。(2)の回路は主電源が完全に切れる前に、データ保持用CMOS・RAMの電源を主電源から、データ保持用電源に自動的に切替える。主電源が断中、コンピュータの動作は勿論停止する。電源が入った場合は、この逆の切替えが自動的に行われる。

また、(2)の回路は主電源が切れた場合に、CMOS・

RAMのCS入力をデータ保持に必要なレベル（ $\overline{CS} \geq V_{DR} - 0.2V$ ）⁽²¹⁾に切替える。ここで、 V_{DR} はデータ保持電源電圧である⁽²¹⁾。主電源が入った場合は、このCMOS・RAMのCS信号は通常通りメモリチップ選択回路の出力CSに切替える。

5. 6 制御グループ

機能 制御グループは本機の機能を達成するために、CPUの制御機能を補う。

構成 制御グループは次の幾つかの制御回路により構成される。タイマやHLT検出回路もこの中に含まれる。図11参照。

- (1) ハードウェアステップ制御（図11の⑤）
- (2) ソフトウェアステップ制御（同上）
- (3) タイマ（図11の⑥）
- (4) HLT検出回路（同⑦）
- (5) 割込み制御（同⑧）

なお、これらの制御回路の幾つかは関連して動作する。例えば、(2)～(4)の回路はそれぞれ割込み要求信号を発生するが、これらの信号は(5)割込み制御回路に入力される。割込み制御回路はこれらの割込み要求入力を管理し、かつCPUに対して割込み要求信号ならびに割込み命令コードを発生する。

以下に各制御回路の機能、構成および仕様について述べる。

5. 6. 1 ハードウェアステップ制御回路

機能 ハードウェアステップ動作の制御を行う。

構成 ハードウェアステップ制御回路の構成は次のとおりである。

- (a) ハードウェアステップ許可F/F
- (b) 命令サイクルカウンタ（最大値mカウンタ*¹）
- (c) マシンサイクルカウンタ（最大値1カウンタまたはウェイト要求F/F）
- (d) その他、ゲート回路等

ハードウェアステップ制御回路の構成図を図22に示す。また、最大値mカウンタの1構成法を図23に示す。

仕様 この制御回路の入力および出力信号の種類、回路全体としての動作仕様ならびに上記(a)～(c)の各構成要素の動作仕様等について述べる。

(1) 入力信号*²

ハードウェアステップ制御回路の入力信号の種類を

* 1 m個の入力パルスを計数したときに出力信号が1となり、しかも(m+1)個以上の入力パルスに対してはこの出力信号1が変化しないような回路のことを本文では最大値mカウンタと呼ぶことにする。

* 2, 3 これらの入力/出力信号の働きについては、(3)以下の仕様説明参照。

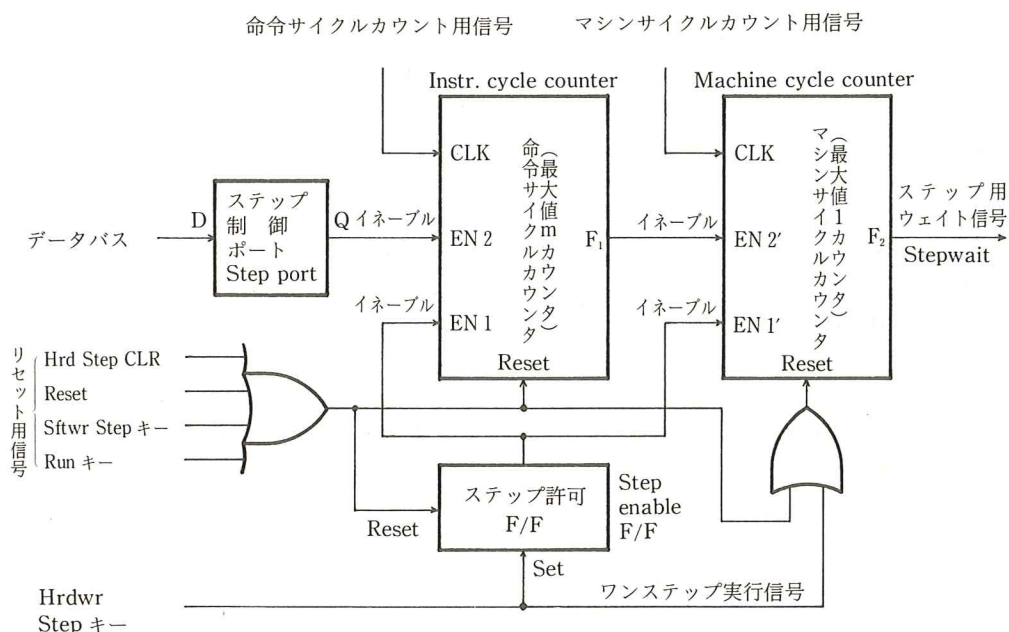


図22 ハードウェアステップ制御回路の構成図

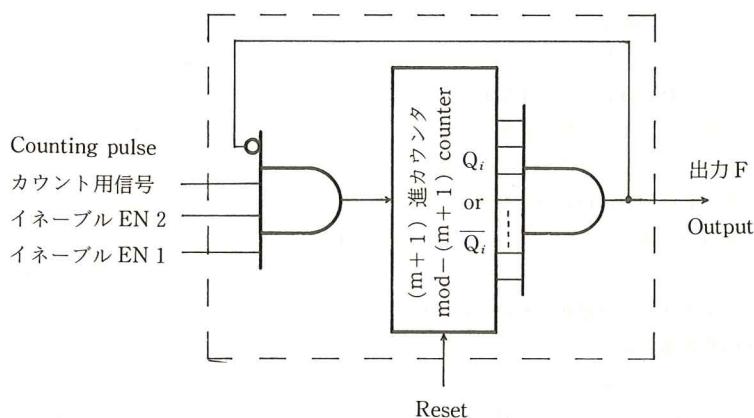


図23 最大値mカウンタの構成例

記す。図22参照。

- (イ) ステップ許可 F/F セット信号 (Hrdwr Step キーより)
- (ロ) 命令サイクルカウンタイネーブル信号 (ハードウェアステップ制御ポートより)
- (ハ) 命令サイクルカウント用信号 (M₁ カウント用信号 STSTB・M₁ (D₅) による)
- (ニ) マシンサイクルカウント用信号 (SYNC 信号

による)

- (ホ) ステップ制御回路のリセット用信号各種 (①プログラムによる Hrd Step CLR 信号, ②システム Reset 信号, ③ Sftwr Step キー, ④ Run キーによる信号)
- (ヘ) ワンステップ実行信号 (Hrdwr Step キーより)
- (2) 出力信号^{*3}

出力信号は1つだけである。

- (ト) ステップ用ウェイト信号 (CPUグループの $\overline{\text{RDYIN}}$ 入力の1つへ)

備考1 ハードウェアステップ動作開始ルーチン

ハードウェアステップ動作を開始するに当たり、モニタプログラムにおいて、ステップ動作開始番地へプログラムの実行を移すための「ステップ動作開始ルーチン」を実行する必要がある。すなわち、ステップ動作開始のための **Hrdwr Step** 入力をモニタにおいて確認すると、モニタは次にこのステップ動作開始ルーチンに入る。このルーチンについては詳しく述べないが、これには少なくとも次の(イ)、(ロ)、(ハ)が含まれなければならない。備考2も参照されたい。

- (イ) 表示プログラム起動用割込み^{*1}の禁止。

これはハードウェアステップ動作中にこの割込みが発生して、目的とするステップ動作プログラムに代りに、表示プログラムがステップ動作で実行されるのを防ぐために必要である。

- (ロ) このステップ制御回路の中の命令サイクルカウンタに対する動作許可 (イネーブル) 信号の発生。
(ハ) ステップ開始番地への飛越し。

ここで、(ハ)は当然ステップ動作開始ルーチンの最後の命令により実行される。このための命令としては8080ACPUの場合、PCHLまたはJMP命令を使える。ただし、本機ではステップ開始番地が可変であるために、このJMP命令 (特にそのアドレス部) はRAMに書く必要がある。PCHL命令を用いる場合は、この命令をROMに書込める。

備考2 ハードウェアステップ動作におけるCPUレジスタの初期値設定

本機の仕様ではこれについて規定していない。ハードウェアステップ動作の開始に当たって、CPUレジスタの初期値設定を可能とするためには、ステップ開始ルーチンの最後、(ハ)ステップ開始番地への飛越しの前に、さらに次のような事項を行う必要がある。すなわち、(イ)CPUレジスタの初期値をメモリから各レジスタに移す。

これにより、ハードウェアステップ動作開始前に(メモリに)CPUレジスタの初期値を設定^{*2}しておけば、ハードウェアステップ動作開始時のCPUレジスタの

内容は、上記の設定値となる。ただし、H、Lレジスタの初期値設定も可能とするためには、(ハ)の飛越し命令にPCHLを使ってはならない。この場合はJMP命令を用いるとよい。

ハードウェアステップ制御回路の全体的動作仕様を次に述べる。図22参照。

(3) 全体的動作仕様

ステップ動作の開始

1. ハードウェアステップキー **Hrdwr Step** を押して、これにより(イ)ステップ許可F/Fをセットする。したがって、図22のEN1およびEN1'が1となる。同時にモニタにより **Hrdwr Step** が押されたのを検出すると、モニタはハードウェアステップ動作開始ルーチンに入り、これにより(ロ)命令サイクルカウンタに対する動作許可信号 (イネーブル信号EN2) をセットする。

2. イネーブル信号EN2がセットされて後、(m+1)個目の命令の第1マシンサイクルにおいて、このステップ制御回路の出力信号、すなわち(ト)ステップ用ウェイト信号により、CPUをウェイト状態にさせる。次項(4)―(ロ)命令サイクルカウンタおよび同(c)マシンサイクルカウンタ参照。

ステップ動作

3. このあと **Hrdwr Step** を押すたびに、ワンステップ実行信号により1マシンサイクルだけ進行させる。すなわち、ワンステップ実行信号によりマシンサイクルカウンタを一瞬リセットすることにより、CPUをウェイト状態から抜け出させてREADY状態にし、マシンサイクルカウンタが次の1マシンサイクルをカウントすると再びステップ用ウェイト信号がセットされて、CPUをウェイト状態にさせる。

4. 2. または 3. のあと **Run**, **Sftwr Step** または **Reset** を押すと、この制御回路がリセットされて、CPUをウェイト状態から直ちに抜け出させる。プログラムによるHrd Step CLR信号を与えた場合も同様である。なお、**Sftwr Step** または **Reset** を押した場合は、プログラムによりステップ制御ポートを通じて、命令サイクルカウンタのイネーブル信号EN2をリセットする。

以上、ハードウェアステップ時にこのステップ制御

* 1 後述表示プログラム起動用タイマの項参照。

* 2 3. 3(ロ)CPUレジスタの項参照。

回路以外からのウェイト要求信号の発生は、ないものと仮定して仕様を述べた。外からのウェイト要求があれば、CPUは上記の記述にかかわらずウェイト状態に入るであろう。

(4) 各構成要素の機能および仕様

ハードウェアステップ制御回路の各構成要素の機能および仕様を次に述べる。

(a) ハードウェアステップ許可 F/F

これはこのステップ制御回路全体、すなわち(b)命令サイクルカウンタおよび(c)マシンサイクルカウンタの動作を許可または禁止する(F/Fの出力 $Q=1$ のとき許可、 $Q=0$ のとき禁止)。上記(1)および(3)で述べたように、このF/Fをセットするのは **Hrdwr Step** による。リセットするのは幾つかのリセット用信号による。図22、(1)の(a)および(3)の4. 参照。

(b) 命令サイクルカウンタ

これはハードウェアステップ動作の開始が、あらかじめ指定された番地から正確に行われるように、ステップ動作開始のタイミングをはかるためのカウンタである。すでに記したように、このカウンタは最大値 m カウンタである。したがって、その動作許可信号EN1が **Hrdwr Step** によりセットされ、つづいてEN2がステップ開始ルーチンの中でセットされると、このカウンタは計数動作を開始し、EN2がセットされた直後より m 個の命令を計数すると、このカウンタの出力 F_1 が1になる。以後、リセット入力に加えられ、カウンタがリセットされるまでこの出力1を保持する。

ここで、 m はステップ開始ルーチンの中のEN2をセットする命令の次から、このルーチンの最後の命令(すなわち、ステップ開始番地への飛越し命令)までの命令数に等しい。逆に、回路の m の値が決まっているならば、上記の命令数も m になるように、ステップ開始ルーチンを作る必要がある。

(c) マシンサイクルカウンタ

マシンサイクルカウンタは最大値1カウンタであり^{*}1、その2つの動作許可信号EN1'およびEN2'がともにセットされているときに、1個のマシンサイクルを計数して、出力 F_2 にステップ用ウェイト信号1を発生する。このあと **Hrdwr Step** によりワンステップ実行信号を入力すると、このカウン

タの出力はリセットされるが、次の1個のマシンサイクルを計数することにより、出力は再び1に戻る。

む す び

本文では教育用基板マイクロコンピュータのハードウェア設計の前半について述べた。後半は続編にゆずる。

参 考 文 献

本文で新しく参照した文献だけを前編⁽¹²⁾の文献に続いて記す。

- (12) 川端：教育用基板マイクロコンピュータの設計(2)、本紀要、22、2 (1981)、pp.121-146。
- (13) Intel Corporation: Component Data Catalog, Intel Corporation (1980)。
- (14) Texas Instruments Incorporated: The Bipolar Digital Integrated Circuits Data Book—First Edition—, テキサスインスツルメンツアジアリミテッド。
- (15) テキサスインスツルメンツ アジア リミテッド：インターフェイスICアプリケーションマニュアルとデータブック、(株)エレクトロニクスダイジェスト (昭52)。
- (16) RAM(編)：マイクロコンピュータ総合カタログ '79、廣済堂出版 (昭54)、pp.68-70。
- (17) たとえば、[8, 9, 13]の中の8257 DMAコントローラの仕様参照。
- (18) [8, 9, 13]の8080A CPUのHOLD信号のタイミング波形および交流特性参照。
- (19) シャープ(株)：Z-80 テクニカルマニュアル、シャープ(株) (1979)。
- (20) 東京芝浦電気(株)：集積回路技術資料 MOSメモリ (第3版)、東京芝浦電気(株) (1980)。
- (21) (株)日立製作所：SEMICONDUCTOR DATA BOOK ICメモリ、(株)日立製作所 (1981)。
- (22) 三菱電機半導体データブック編集委員会(編)：'82三菱半導体データブック マイクロコンピュータ関連LSI編、三菱電機(株)半導体事業部 (昭57)。

(つづく)

* 1 このカウンタを最大値 n カウンタにすれば、 n マシンサイクルを1ステップとするステップ動作を行わせることができよう。ただし、この場合ステップ開始の正確なタイミングについてはさらに工夫を要する。